FinTech Unit 2 Homework PyBank: Grading Rubric

Criteria	Ratings				Points	Feedback
Set up Project • GitHub repo created - named: python-homework. • Directory created with appropriate activity. • **main.ipynb** added to main folder. • Repo updated with latest changes.	**35 Points Mastery** • Completed 4 out of 4 requirements • Code Runs without error and produces the assigned results • Code is creatively engineered where possible • Code uses concise logic	**34 > 28 Points Approaching Mastery** • Completed 3 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenerio • Code is free of bugs	**28 > 23 Points Progressing** • Completed 2 out of 4 requirements • Code runs without error • Code produces results as expected 80% of the time	**23 > 0 Emerging** • Completed 1 or fewer out of the 4 requirements • No submission • Code runs with error	0	
Analysis and Calculations • Total number of months in dataset included. • Net total amount of Profit/Losses calculated over entire period. • Average of the changes in Profit/Losses calculated over entire period. • Greatest increase in Profits calculated over entire period (Date and Amount). • Greatest decrease in Losses calculated over entire period (Date and Amount). • Analysis printed and text file exported with final results.	**35 Points Mastery** • Completed 6 out of 6 requirements • Code Runs without error and produces the assigned results • Code is creatively engineered where possible • Code uses concise logic	**34 > 28 Points Approaching Mastery** • Completed 5 out of the 6 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenerio • Code is free of bugs	**28 > 23 Points Progressing** • Completed 4 out of the 6 requirements • Code runs without error • Code produces results as expected 80% of the time	**23 > 0 Emerging** • Completed 3 or fewer out of the 6 requirements • No submission • Code runs with error	0	
Coding Conventions/Formating • Appropriate header, name, short description at top of the notebook • Imports are at the top of the file, just after any headers or subheads. • Files read in from relative file path • Functions and variable names are descriptive, lowercase, with words separated by underscores • Clean code, no repetition, maintainable and highly reusable code. • Appropriate code wrapping and cell sizes • Appropriate subheads as needed	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**	0	
Deployment/Submission • Files submitted in personal repo • Appropriate directory structure with correct files needed to run scripts • Appropriate commit messages • Appropriate README	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**	0	
Documentation/Comments • Code is well commented with concise, relevant comments	**10 Points Mastery**	**9 Points Approaching Mastery**	**8 Points Progressing**	**8 > 0 Emerging**	0	

LETTER GRADE

TOTAL POINTS 0   F